

Codezero移植工作分工

柯杰伟

June 17, 2014

1 目标

总的目标是，通过对codezero源代码的分析为基础，移植codezero到powerpc架构的P4080DS八核机器上，并进一步研究多核确定性和运行时间估算的课题。另一方面，结合Arinc 653对分区隔离和通信的基准，在codezero中加入这些新的特性。

2 任务划分

下面列出如今划分的四项任务，并分配到每位同学，请大家注意任务进行的同时，形成中间过程文档，以及最后的成果。

2.1 分析scons脚本（禹舟健）

scons是类似于Makefile，用于指导项目编译的管理文件，使用Python语言编写。位于scripts目录中。

我们在进行移植时候，使用的是powerpc的工具链。

了解codezero编译流程，主要有几点：找到编译使用的工具和选项，我们需要替换工具的定义；找到生成中间文件和最终目标文件final.elf的流程；了解编译依赖关系，可以通过查看编译时打印信息来直观了解。

2.2 分析loader源代码，以便进行ppc的移植。（柯杰伟）

loader的代码比较简单，主要是进行较少的初始化，以及加载final.elf映像。

对比使用loader和u-boot的工作量和好处。考虑使用loader源代码进行加载，还是使用P4080DS的U-boot程序进行加载。u-boot改动工作量较大，之前编译生成的u-boot在P4080DS运行时，网卡驱动还没有尝试加进去，可以先进行尝试，以便通过ftp加载系统映像运行。

2.3 分析src/include的内核源代码（康乔，柯杰伟，zheng si ting）

了解内核的实现，找到体系结构和平台相关的代码部分（arch/platform），以便进行ppc的移植。

先从运行时入手，了解系统从loader加载完内核开始，到开始运行container程序的流程。

然后再依次分析每个文件。

其中，包括了以下目录：

src/api 向container提供的api功能，例如cache，irq，线程，系统调用等

src/arch 包含了体系结构（arm）相关的.s代码，例如硬件初始化head.s，系统调用syscall.s等

src/drivers 包含timer，irq，uart三种类型驱动

src/generic 通用的内核功能，例如调度器，tcb，container，resource管理，时钟等

src/glue 包含了体系结构（arm）相关的.c代码，例如cache，irq，内存等

src/lib 一些C库

src/platform 硬件平台相关的代码

2.4 了解container源代码，运行Linux的案例（张健）

我们希望能够在codezero上运行其他的操作系统。由于codezero提供了在container运行linux的编译脚本，可能运行linux的可能性较大。

codezero提供了在container运行linux的编译脚本，并指定了linux的版本号，使用对应版本号的标准linux内核编译会报缺少头文件的错误。可见，codezero运行linux应该对标准linux做了修改，可以参考OKL4的实现。

由于OKL4和codezero提供的底层接口可能有较大差异，在上面跑linux可能有一定难度，可以考虑运行其他os。